Volume 0 (1981), Number 0 pp. 1–10 COMPUTER GRAPHICS *forum*

Test-Time Optimization for Video Depth Estimation Using Pseudo Reference Depth

Libing Zeng and Nima Khademi Kalantari

Texas A&M University, United States

Abstract

In this paper, we propose a learning-based test-time optimization approach for reconstructing geometrically consistent depth maps from a monocular video. Specifically, we optimize an existing single image depth estimation network on the test example at hand. We do so by introducing pseudo reference depth maps which are computed based on the observation that the optical flow displacement for an image pair should be consistent with the displacement obtained by depth-reprojection. Additionally, we discard inaccurate pseudo reference depth maps using a simple median strategy and propose a way to compute a confidence map for the reference depth. We use our pseudo reference depth and the confidence map to formulate a loss function for performing the test-time optimization in an efficient and effective manner. We compare our approach against the state-of-the-art methods on various scenes both visually and numerically. Our approach is on average 2.5X faster than the state of the art and produces depth maps with higher quality.

CCS Concepts

• ***Computing methodologies*** → *Depth Estimation, Optimization;*

1. Introduction

Depth estimation from a video sequence plays an important role in many computer graphics and vision applications, such as view synthesis [TS20, ZTF*18, PZ17], video stabilization [LGJA09], scene understanding [GGAM14, QLW*17, SLX15], special video effects [LHS*20], and augmented reality [VKB*18]. However, accurate and consistent depth estimation from casually captured videos is still challenging because of higher noise level, motion blur, and rolling shutter deformations. With decades of studies in this field, a great number of techniques have been developed to approach depth estimation from videos.

Recently, several hybrid approaches [LHS*20,KRH21,ZCT*21] propose to combine the strength of the learning-based and traditional techniques. These approaches use a pre-trained single image depth estimation network and fine-tune its weights through a test-time optimization process. By doing so, the network learns to satisfy the geometry of the scene through test-time optimization, while relying on the learned priors in the regions with weak constraints. Unfortunately, the test-time loss requires forward and backward evaluation of a complex reprojection process. Furthermore, the loss in these approaches equally enforces all the geometric constraints, even the inaccurate ones. As a results, these methods are computationally expensive and produce sub-optimal results in challenging cases, as shown in Fig. 8.

We address these issues by proposing a novel test-time loss function. The key contribution of our work is to compute pseudo reference depth maps for each frame by analytically minimizing the distance between optical flow and depth-reprojected correspondences. Furthermore, we propose a simple strategy using the median operation to discard the inaccurate depth maps and generate a single pseudo reference depth at each frame. The pseudo reference depth maps are essentially optimal depth at each frame according to pairwise optical flows. Since the optical flows are not always accurate, the estimated pseudo reference depth maps in certain regions may be unreliable. Therefore, we also propose a method to compute a confidence map indicating the reliability of the computed pseudo reference depth. During the test-time optimization, we enforce the estimated depth using the neural network to be similar to the pseudo reference depth according to the confidence map. Moreover, we ensure the consistency of the estimated depth in the neighboring frames by enforcing the 3D projected corresponding pixels in the consecutive frames to be similar.

We demonstrate the superiority of our method through extensive comparisons against the state-of-the-art algorithms on a variety of scenes, including publicly available datasets such as TUM RGB-D [SBC12], KITTI [GLU12], and NYU Depth [NSF12]. Our approach is visually and numerically better than the state of the art and is on average 2.5X faster than Luo et al.'s method [LHS*20]. In summary, we make the following contributions:

- We introduce "pseudo" reference depth maps to accelerate the test-time optimization process.
- We propose a method to discard inaccurate pseudo depth maps

along with a confidence map to ensure only the consistent geometric constraints are used during optimization.
- We demonstrate that our approach outperforms the state of the art on several datasets both numerically and visually.

2. Related Work

Single Image Depth Estimation. In recent years, significant progress has been made on supervised learning-based single image depth estimation [EPF15, LSL15, LRB*16, EF14, FGW*18]. However, the major challenge is acquiring diverse images with their corresponding ground truth depth maps for training the networks. Many approaches address this issue through the use of synthetic datasets [MIH*16], relative depth annotations [CFYD16], 3D movies [RBK21, WLPW19], and depth maps obtained by structure-from-motion and multi-view stereo [LS18, CQD19, LDC*19]. Another set of methods propose to address this problem through self-supervised loss functions using monocular videos [DPH*20, QLL*18, QLL*20, RJB*19, VRS*17, YS18, ZBSL17, ZLH18] or stereo pairs [GLY*18, GMB17, GLJA19]. However, the major problem with all of these approaches is that they are specifically designed for single image depth estimation, and generate results with severe temporal flickering on videos.

Video Depth Estimation. A couple of approaches [LGK*19, ZJ20] propose to estimate depth of a video by training a network using multi-view loss functions. However, these methods are inherently designed for static scenes. Several approaches [PGDG20, WPF19, ZSL*19] propose to implicitly enforce temporal coherency through recurrent neural networks. Yoon et al. [YKG*20] propose to fuse the estimated depth from a single image and multi-view stereo through a learned combiner module. However, these approaches do not explicitly enforce the final depth maps to be geometrically consistent. Finally, Li et al. [LLZ*21] propose to fine-tune a single-image depth estimation network using unlabelled video dataset by enforcing a set of geometric and temporal constraints. While this approach produces reasonable results for videos that are similar to their fine-tuning dataset, it often struggles to generalize to unseen videos.

Test-time Optimization. A few recent techniques propose to generalize the neural radiance field method [MST*20] to dynamic scenes [LNSW21, PSB*21]. While these approaches aim to synthesize novel views, their estimated opacity field can be used to reconstruct the depth at each frame. Unfortunately, these approaches are extremely expensive and can only be used for short video sequences. A couple of methods [CPMA19, CQD19] propose to fine-tune a single image depth estimation network on the test scene. However, these approaches focus on increasing the quality of the single image depth. To ensure the estimated depth maps are geometrically consistent, Luo et al. [LHS*20] propose a test-time loss based on global geometric constraints. Their approach, however, has two major issues. First, they require forward and backward evaluation of a complex loss requiring pixel reprojection. Moreover, they assume all the geometric constraints are equally reliable and ignore the inaccuracies of the optical flows. Therefore, their method is computationally expensive and produces sub-optimal results in challenging cases. Kopf et al. [KRH21] and Zhang et al. [ZCT*21] extend Luo et al.'s method [LHS*20] to optimize the camera poses and improve its performance on scenes with large

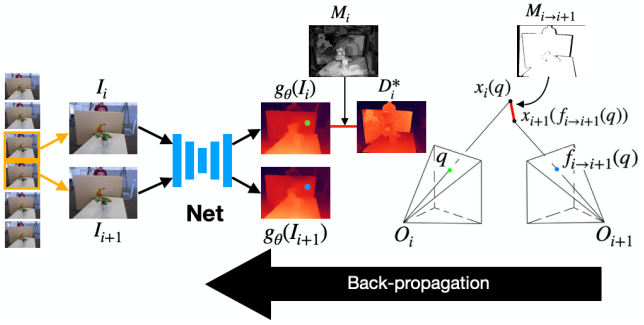

Figure 1: *Given an input video, we fine-tune the parameters of an existing single image depth estimation network, g_θ, to produce consistent depth estimates. We do so by minimizing an objective function, consisting of two loss terms, shown with the red lines. Specifically, we compute a pseudo reference depth map for each frame D_i^* (discussed in Sec. 3.2) and minimize the error between the estimated and pseudo reference depth maps. Note that, we weight this error using a confidence map M_i to suppress the loss in regions where the pseudo reference is unreliable. Additionally, we use the estimated depth maps to project the corresponding pixels (shown with green and blue circles) in a pair of consecutive frames into 3D and minimize the loss between the projected 3D points. We suppress this loss in the occluded regions using a mask $M_{i\to i+1}$, computed through optical flow forward backward consistency test.*

motion, respectively. However, they have the same drawbacks of Luo et al.'s method [LHS*20] since they utilize the same major test-time loss functions.

3. Algorithm

Given a set of N frames, $I_1, \cdots, I_N$, of a monocular video, we estimate geometrically and temporally consistent depth for each frame. We do this by first obtaining an initial depth estimate from each frame using an existing deep single image depth estimation network, $g_\theta(I_i)$. Our goal is then to improve these depth estimates by updating the network parameters θ through a novel test-time optimization process. We show the overview of our algorithm in Fig. 1. In the following sections, we first describe the pre-processing step and then discuss our test-time optimization process.

3.1. Pre-processing

We pre-process the data by following the proposed strategy by Luo et al. [LHS*20]. Specifically, we first use COLMAP [SF16, SZPF16], an off-the-shelf structure-from-motion (SfM) software, to obtain the camera poses. These parameters are later used to obtain our geometric losses. Furthermore, we use an existing single image depth estimation method (Li et al. [LDC*19] or Godard et al. [GAFB19]) to obtain an initial depth for each frame. To fix the scale mismatch between these depth estimates and SfM, we scale the camera translation of each frame using the average of the median of the ratio between the learning-based depth estimates and the ones from SfM. Moreover, we compute forward and backward flows, using FlowNet2 [IMS*17], between frames I_i and I_j where the neighboring frames are selected according to the power

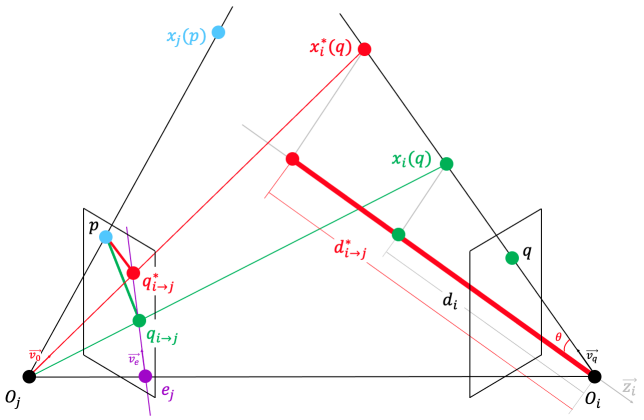

Figure 2: *Using an initial depth d_i, the pixel q from camera i can be projected into the 3D world ($x_i(q)$) and back into camera j to obtain the reprojected pixel $q_{i\to j}$. This reprojected pixel always lies on the epipolar line shown in purple. The distance between this reprojected pixel $q_{i\to j}$ and the optical flow correspondence p indicates how well the depth and optical flow match. The pseudo reference depth $d^*_{i\to j}$ is the one resulting in smallest distance.*

of 2 rule, proposed by Luo et al. [LHS*20]. For each pair (i, j), we use the forward backward consistency to obtain a binary mask $M_{i\to j}$ representing the occluded areas and regions with inconsistent flows.

3.2. Pseudo Reference Depth

The main challenge for an optimization system is designing an appropriate loss function. Our key contribution is to compute a "pseudo" reference depth map with its corresponding confidence map for each frame. During test-time optimization, we minimize the confidence weighted distance between the network's output and the pseudo reference depth. Additionally, we introduce a loss to ensure the estimated depth maps in neighboring frames are consistent. In summary, our objective consists of the following two terms:

$$\mathcal{L} = \mathcal{L}^*_{\text{pseudo}} + \lambda \mathcal{L}_{\text{cons}} \tag{1}$$

where $\mathcal{L}^*_{\text{pseudo}}$ and $\mathcal{L}_{\text{cons}}$ are defined in Eqs. 8 and 9, respectively. Moreover, λ is a parameter that controls the balance between the two terms. We set this parameter to 0.3 in all the experiments, unless otherwise stated. Next, we explain these two terms in detail.

Per-pair Pseudo Reference Depth. The goal here is to obtain a pseudo reference depth for frame i considering frame j. To do this, we rely on the observation that the displacement computed using optical flow between a pair of images should match the displacement obtained by depth reprojection. Specifically, let pixel q in camera i correspond to pixel p in camera j, as shown in Fig. 2. Using the depth, d_i, we can project pixel q in the 3D world and back to camera j to obtain the depth reprojected pixel $q_{i\to j}$. The distance between pixel p and the reprojected pixel q, known as spatial loss in previous approaches [LHS*20, KRH21, ZCT*21], indicates how well the depth d_i matches the optical flow. We then obtain the pseudo reference depth $d^*_{i\to j}$ in camera i considering camera j by minimizing this distance as follows:

$$d^*_{i\to j} = \arg\min_{d_i} \|p - q_{i\to j}\|, \tag{2}$$

where $q_{i\to j}$ depends on d_i and the optimal d_i corresponds to the reprojected pixel $q_{i\to j}$ that is closest to pixel p.

Fortunately, the optimal solution to this objective function can be analytically computed as follows (see the Appendix for derivation):

$$d^*_{i\to j} = \frac{(o_j - o_i)\cdot \vec{v_q} - (\vec{v_o}\cdot\vec{v_q})\,(o_j - o_i)\cdot\vec{v_o}}{1-(\vec{v_q}\cdot\vec{v_o})^2}\,\|cos(\theta)\|, \tag{3}$$

where o_i and o_j are the center of projection of cameras i and j, respectively. Furthermore, $\vec{v_q}$ is the unit vector that points to pixel q from o_i (see Fig. 2). Moreover, $\vec{v_e}$ is a unit vector defined as:

$$\vec{v_o} = \text{norm}(e_j - o_i + ((p - e_j)\cdot\vec{v_e})\cdot\vec{v_e}), \tag{4}$$

where e_j is the epipole corresponding to pixel q on camera j and $\vec{v_o}$ is the unit vector defining the direction of the epipolar line. Furthermore, norm is the vector normalization operator. See Fig. 3 (bottom) for examples of per-pair pseudo reference depth maps.

Note that for accurate optical flow and camera calibration, the line from the center of camera j (o_j) to pixel p intersects with the line connecting the center of camera i (o_i) and pixel q. In this case, the projected distance between this intersection point and o_i is the "ground truth" depth at pixel q. However, in practice, optical flow and camera calibration are not accurate, and thus the two lines do not intersect. Therefore, we define the depth minimizing the objective in Eq. 3 as the "pseudo" reference depth for pixel q.

Using this approach, we can compute a set of pseudo reference depth maps for frame i considering a set of neighboring frames j. We can then define the pseudo reference loss in Eq. 1 as the weighted summation of the loss between the estimated depth by the network and all the pseudo reference depth maps as follows:

$$\mathcal{L}_{\text{pseudo}} = \sum_i \frac{\sum_j M_{i\to j}\mathcal{L}(g_\theta(I_i), D^*_{i\to j})}{\sum_j M_{i\to j}}, \tag{5}$$

where $M_{i\to j}$ is a binary mask indicating the regions with inconsistent flows based on the forward backward flow test. Moreover, $g_\theta(I_i)$ is the estimated depth map at frame i using the single image depth estimation network. Finally, $D^*_{i\to j}$ is the pseudo reference depth map at frame i considering frame j ($d^*_{i\to j}$ for all pixels).

The major problem with this loss is that it ignores the inaccuracies of the optical flow and considers the pseudo reference depth from all the frame pairs (i, j) as ground truth. Note that even though we mask out the regions with inconsistent forward and backward flows using $M_{i\to j}$, many inaccurate flows would still pass this consistency test. Therefore, the pseudo reference depth maps computed using these flows are consequently inaccurate. Next, we address this problem by proposing a simple, but effective strategy.

Discussion – Existing techniques [LHS*20, KRH21, ZCT*21] also utilize the distance between the depth reprojected pixel q and pixel p, but they directly use it along with other loss functions to optimize the network parameters. Our approach of computing pseudo reference depth maps has two main advantages. First, our approach is more efficient as we compute the pseudo reference depth maps

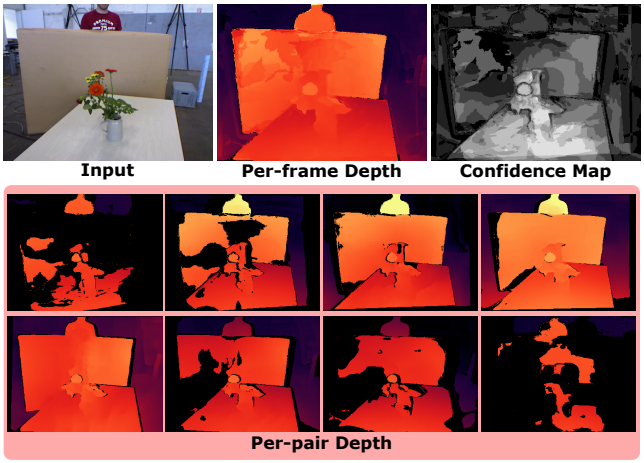

Figure 3: *On the bottom, we show a set of per-pair pseudo reference depth for the input image (top left) considering a set of neighboring frames. Because of the inaccuracies of the optical flow, these per-pair pseudo reference depth are not always accurate. For example, the depth for the person standing behind the board is incorrectly estimated to be close to the camera in some of the depth maps. By computing the median of these per-pair depths, we discard the inaccurate depth values and generate a more accurate per-frame pseudo reference depth (top middle). The confidence map (top right) indicates the reliability of the per-frame pseudo reference depth.*

by analytically optimizing Eq. 3 prior to optimization. In contrast, the existing methods require forward and backward (for gradients) evaluation of the distance in Eq. 3 in every iteration of the optimization. Second, by directly using this distance in optimization, existing methods ignore the inaccuracies of the optical flows. However, as discussed below, computing the pseudo reference depth allows us to discard the depth maps computed with inaccurate flows.

Per-frame Pseudo Reference Depth. As discussed, the inaccuracies in the pseudo reference depth maps negatively impacts the quality of the results. To mitigate this issues, we make an observation that all the computed per-pair pseudo reference depth maps for a particular frame ($D^*_{i\to j}$ where j is the index of a set of neighboring frames) should have similar the depth values. Therefore, the ones that do not agree with the majority are computed using inaccurate flows and should not be taken into account during optimization. Based on this observation, we propose to compute per-frame pseudo reference depth by obtaining the median of all the per-pair pseudo reference depth maps at each frame as follows:

$$D^*_i = \underset{j}{\text{median}}(D^*_{i\to j}) \tag{6}$$

Note that the median operation is robust to outliers, and thus the depth values computed from inaccurate flows are automatically discarded with this simple strategy, as shown in Fig. 3.

This per-frame pseudo reference depth D^*_i can be used as the target to optimize the single image depth estimation network. However, every depth value in D^*_i is not equally reliable. For example, the values that are in agreement with all the per-pair pseudo reference depth maps are more reliable than the ones that only match a few per-pair depth maps. To account for this, we compute a confidence map by summing the binary mask $M_{i\to j}$ for the per-pair depth maps that are in agreement with the per-frame depth as:

$$M_i = \sum_j s_j M_{i\to j},\ s_j = \begin{cases} 1 & \|D^*_i - D^*_{i\to j}\| \le 0.1 D^*_i \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

We use this confidence map M_i along with the per-frame reference depth D^*_i to formalize our pseudo loss.

3.3. Test-Time Loss

Pseudo Reference Loss. We define our final pseudo reference loss as the weighted distance between the estimated and pseudo reference depth maps:

$$\mathcal{L}^*_{\text{pseudo}} = \sum_i M_i \| \log(1 + g_\theta(I_i)) - \log(1 + D^*_i) \|. \tag{8}$$

Note that we compute the loss in the logarithmic domain to put more emphasis on the smaller depth values.

Consistency Loss. The loss in Eq. 8 is applied to each individual frames, and thus, does not guarantee that the estimated depth at different frames are consistent. To ensure such a consistency, we enforce the 3D points from the corresponding pixels of two consecutive frames to be similar. Specifically, we do this by minimizing the following loss:

$$\mathcal{L}_{\text{cons}} = \sum_q \sum_i M_{i\to i+1} \| x_i(q) - x_{i+1}(f_{i\to i+1}(q)) \|, \tag{9}$$

where $x_i(q)$ is the projected pixel q in frame i to the 3D world (see Fig. 2) and $f_{i\to i+1}(q)$ is the pixel in frame $i+1$ that corresponds to pixel q in frame i according to the optical flow.

Note that existing techniques [LHS*20, KRH21, ZCT*21] use a similar loss, called disparity loss, to enforce consistency between neighboring frames. However, their loss only enforces the similarity of the projected 3D points across one dimension. In contrast, our consistency loss considers all the three dimensions when enforcing the similarity of the projected 3D points.

4. Results

We implement our approach in PyTorch and use ADAM [KB15] to perform the test-time optimization with the default parameters. We fine-tune the network for 15 epochs with a batch size of 3 and a learning rate of 3×10^{-5} to produce all the results, unless otherwise stated. Throughout this section, we compare our approach against the base single image depth estimation methods of Li et al. [LDC*19] (MC) or Clément et al. [GAFB19] (Monodepth2), as well as the video depth estimation methods of Wang et al. [WLPW19] (WSVD), Luo et al. [LHS*20] (CVD), Kopf et al. [KRH21] (RCVD), and Li et al. [LLZ*21] (TCM). For all the approaches, we use the source code provided by the authors. We extensively evaluate our approach numerically on three publicly available datasets: TUM RGB-D [SBC12], NYU Depth [NSF12], and KITTI [GLU12]. We also provide visual comparisons on scenes

Table 1: *Quantitative comparisons against the other approaches on the TUM RGB-D dataset.*

	Error Metric ↓				Accuracy Metric ↑		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$
MC [LDC*19]	0.3112	0.1038	0.2461	0.3898	0.5671	0.7680	0.8814
WSVD [WLPW19]	0.2923	0.0900	0.2373	0.3796	0.5387	0.7819	0.8939
CVD [LHS*20]	0.1455	0.0587	0.1441	0.2142	0.8173	0.9323	0.9690
RCVD [KRH21]	0.1723	0.0658	0.1524	0.2418	0.7824	0.8733	0.9277
TCM [LLZ*21]	0.2514	0.0804	0.2172	0.3305	0.5432	0.7703	0.9026
Ours	**0.1339**	**0.0335**	**0.1222**	**0.1872**	**0.8262**	**0.9394**	**0.9809**

Table 2: *Quantitative comparisons against the other approaches on the NYU Depth Dataset V2.*

	Error Metric ↓				Accuracy Metric ↑		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$
MC [LDC*19]	0.1920	0.0311	0.1235	0.2484	0.6825	0.9101	0.9744
WSVD [WLPW19]	0.2207	0.0383	0.1309	0.2914	0.6155	0.8701	0.9540
CVD [LHS*20]	0.1875	0.0274	0.1233	0.2289	0.7003	0.9136	0.9758
RCVD [KRH21]	0.1884	0.0263	0.1345	0.2213	0.7045	0.9188	0.9790
TCM [LLZ*21]	0.2041	0.0329	0.1252	0.2529	0.6033	0.8942	0.9687
Ours	**0.1795**	**0.0232**	**0.1159**	**0.2034**	**0.7184**	**0.9279**	**0.9845**

from these three datasets, as well as several casually captured dynamic scenes. For all the optimization-based methods (CVD, RCVD, and ours), we use the same approach as the base single image depth estimation network.

Numerical Comparisons. We first show quantitative comparison against the other approaches on the TUM RGB-D dataset [SBC12] for the 11 scenes in the "3D Object Reconstruction" category. We strictly follow the protocol presented by Luo et al. [LHS*20] for these comparisons. Specifically, we use the ground truth camera poses provided by the dataset for CVD and our approach. RCVD optimizes the camera poses, while the remaining methods do not perform test-time optimization, and thus do not use the ground truth camera poses. We use every other 5 frames of each sequence and resize the images so the longer dimension has a resolution of 384. Here, we use the approach by Li et al. [LDC*19] (MC) as the base single image depth estimation network. Because of the scale ambiguity, we align all the generated depth maps to the ground truth using per-image median scaling. We evaluate the errors in the disparity space and report the average error across various metrics in Table 1. As seen, our approach is considerably better than all the other methods across all the metrics. Note that we fine-tune CVD for the recommended 20 epochs, while using only 15 epochs for our approach as it converges faster.

Next, we evaluate our method on the NYU Depth Dataset V2 [NSF12] using the 694 frames from 234 test scenes based on the official train/test split. We downsample each sequence to around 200 frames and calculate the camera poses of these frames using COLMAP [SF16, SZPF16]. As the base network, we use the single image depth estimation method of Li et al. [LDC*19] (MC). We compute the average errors of all 694 test frames in the disparity space. As shown in Table 2, our method outperform the other algorithms across all the metrics.

We also numerically evaluate our approach on the KITTI dataset [GLU12] by utilizing the Eigen test split [EF14]. Following Luo et al. [LHS*20], we use COLMAP [SF16, SZPF16] to estimate the camera poses for the test sequences. Moreover, we use the FlowNet2 model, fine-tuned with KITTI training sets (FlowNet2-ft-kitti), to generate dense correspondence between frame pairs. Following Luo et al.'s approach [LHS*20], we only sample frame pairs with larger than 50% forward-backward flow consistency. Moreover, to be consistent with Luo et al.'s evaluation protocol, we utilize Monodepth2 [GAFB19] as the base network on this dataset. Note that we use the same base network for all the optimization-based methods (CVD, RCVD, and ours). We fine-tune the network with $\lambda = 1$ and a learning rate of 4×10^{-5}. As shown in Table 3, our results are better than WSVD [WLPW19], CVD [LHS*20], RCVD [KRH21] and TCM [LLZ*21] across all the metrics. Compared to the base single image system [GAFB19], our approach produces comparable results. However, as shown in Fig. 4, our approach outperforms Monodepth2 in about 90% of the test frames.

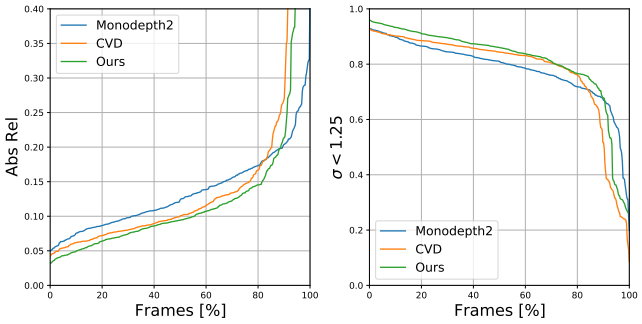

Figure 4: *Quantitative comparison of Monodepth2, CVD, and Our approach on Eigen test split of the KITTI dataset. We illustrate the sorted absolute relative error and accuracy ($\sigma < 1.25$) metrics for all testing frames. Our approach outperforms other methods in about 90% of the test frames.*

Table 3: *Quantitative comparisons against the other approaches on the KITTI dataset.*

	Error Metric ↓				Accuracy Metric ↑		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$
Monodepth2 [GAFB19]	**0.1382**	**0.9714**	5.1642	0.2232	0.8403	**0.9432**	**0.9756**
WSVD [WLPW19]	0.1579	1.9890	5.3272	0.2481	0.8024	0.9143	0.9577
CVD [LHS*20]	0.1501	1.8954	5.2192	0.2358	0.8365	0.9253	0.9665
RCVD [KRH21]	0.1483	1.732	5.2037	0.2380	0.8336	0.9187	0.9604
TCM [LLZ*21]	0.1496	1.8448	5.1753	0.2421	0.8383	0.9295	0.9642
Ours	0.1443	1.2543	**4.9061**	**0.2034**	**0.8517**	0.9393	0.9740

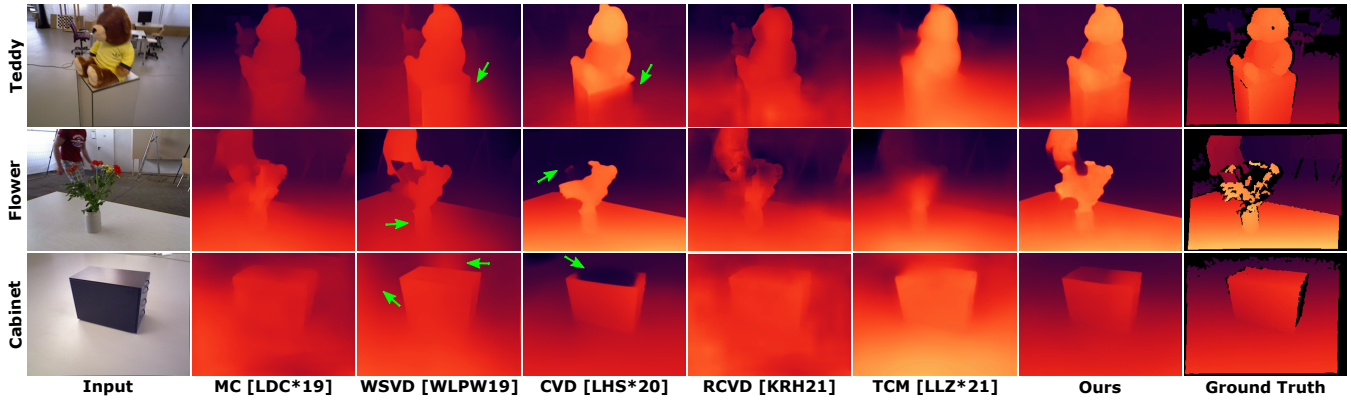

Figure 5: *Comparisons against several state-of-the-art methods on three scenes from the TUM RGB-D dataset*

Visual Comparisons. We begin by comparing our approach against the other methods on three scenes from the TUM RGB-D dataset in Fig. 5. While MC [LDC*19] and WSVD [WLPW19] are able to distinguish different objects, their estimated depth maps are not geometrically consistent and do not match the ground truth as they are not globally optimized. In contrast, CVD [LHS*20] performs test-time optimization (similar to ours) and produces consistent depth maps. However, they produce sub-optimal results in challenging cases as all the geometric constraints (even the inaccurate ones) equally contribute to their test-time loss. For example, their method is not able to properly reconstruct the person in the Flower scene and the top of the cabinet in the Cabinet scene. Our formulation discards the depth values generated based on inaccurate geometric constraints and produces considerably better results in these regions. RCVD [KRH21] builds on CVD's formulation and additionally optimizes the camera poses, but struggles to properly estimate the poses on these textureless scenes, producing results with severe artifacts. Finally, TCM [LLZ*21] produces oversmoothed results as the fine-tuned network is not able to generalize to these scenes. We also show visual comparisons on several scenes from the NYU Depth and the KITTI datasets in Figs. 6 and 7, respectively. Overall, our reconstructed depth maps are more consistent, contain finer details, and better match the ground truth. In particular, note that CVD has difficulty handling the scenes in the KITTI dataset because of their significant motion.

Next, we show comparisons on four casually captured dynamic scenes in Fig. 8. The full videos are provided in the supplementary video. We capture the two scenes at the bottom, but the scenes at the top are from CVD [LHS*20]. Note that the ground truth depth maps for these scenes are not available. Overall, our approach is able to estimate depth maps with higher quality and sharper features in both static and dynamic regions. For example, other methods have difficulty estimating the boundaries of the moving dog (Dog scene), hand (Waving scene), and Jam (Jam scene).

Timing. We evaluate the timings on a GeForce RTX 2080 Ti GPU with 11 GB of memory for the scenes in TUM RGB-D dataset with images of resolution 384×244 (or 244×384). On average our method takes around 4.6 seconds (1.1s for computing the pseudo reference depth and 3.5s for optimization) to generate the depth map for each frame. In comparison, CVD [LHS*20] is 2.5X slower and takes around 11.6 seconds per frame. Our speed up slightly depends on the number of pairs used during the optimization, where larger number of pairs results in more speed up. In the best case scenario, we achieve a 2.8X speed up, while our speed up in the worst case scenario is around 1.9X.

4.1. Ablation Study

We perform several experiments to evaluate the importance of the design choices in our algorithm. For these experiments, we numerically evaluate the results on the TUM RGB-D dataset [SBC12] and report the values in Table 4. We first evaluate the effect of each loss in our system, i.e., the pseudo reference and consistency losses. As seen, both of these losses are important to achieve high-quality results. Specifically, the pseudo loss is the more important term as it encodes the geometric constraints into the pseudo reference depth maps. We also experimented with replacing our consistency loss with the disparity loss from CVD [LHS*20]. As seen, our full approach using the consistency loss produces considerably better results than the one using the disparity loss. This is mainly because the disparity loss only minimizes the distance between the 3D points across one dimension, while the consistency loss considers all the three dimensions.

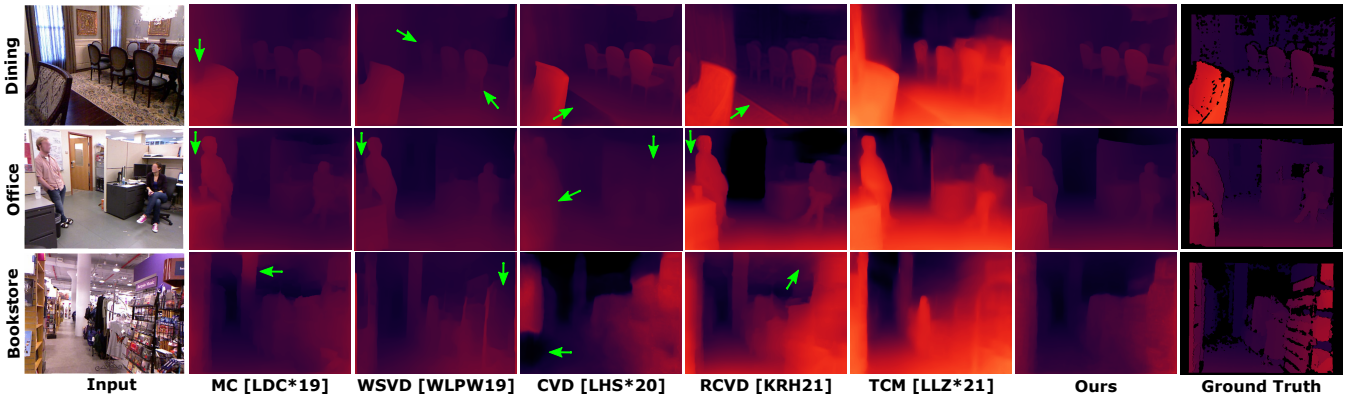

Figure 6: *Comparisons against several state-of-the-art methods on three scenes from the NYU Depth Dataset V2.*

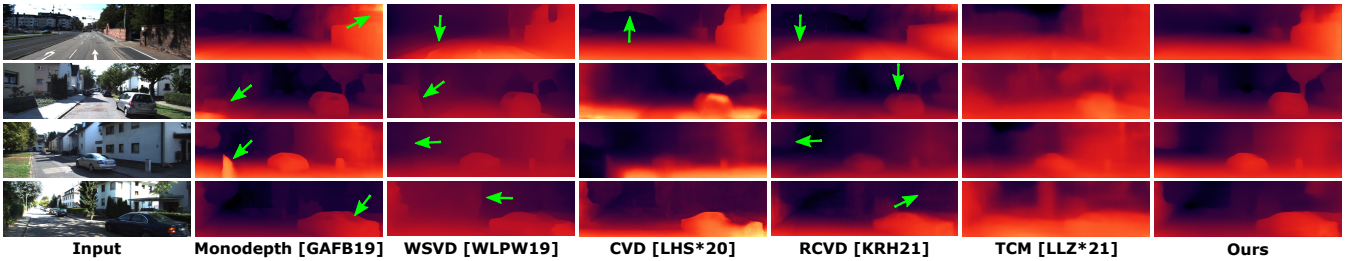

Figure 7: *Comparisons against several state-of-the-art methods on four scenes from the KITTI Dataset. Both WSVD and Monodepth struggle to correctly estimate the depth of the shadowed areas and reflective materials. Moreover, CVD is not able to properly handle these scenes because they typically contain large moving objects. RCVD performs slightly better than CVD, but is not able to properly handle reflective surfaces. Moreover, TCM produces blurry depth maps. In the contrast, our approach handles shadowed areas, reflective materials, and the moving objects with a reasonable accuracy.*

Next, we show the effectiveness of our approach for generating the per-frame pseudo reference depth and confidence maps. Specifically, without the median operation and using Eq. 5 as the pseudo reference loss, our method produces considerably worse results demonstrating the effectiveness of the median operation in discarding the inaccurate depth maps.

Then, we show the effect of our test-time optimization process in Fig. 9. Here, we compare the estimated per-frame pseudo depth maps with the result of our system after test-time optimization on three scenes. As shown, our optimization is highly effective in improving the inaccuracies in the per-frame psuedo depth maps specially in the dynamic regions.

Finally, we evaluate the effect of λ in Eq. 1 by setting extra experiments on TUM RGB-D dataset [SBC12] with λ as 0.1 and 0.9 respectively. As shown in Table 4, smaller or larger weight of consistency loss in our objective function would generate inferior numerical results. Moreover, we also evaluate it on casually captured dynamic Waving scene, the corresponding visual results show that the smaller λ, 0.1, introduces flickering artifacts in the estimated video depths while the larger λ, 0.9, generates over-smoothed results, which are shown at the end of our supplementary video.

4.2. Limitations and Future Work

Although we demonstrated that our approach produces high-quality depth maps, it has several limitations. First, similar to CVD [LHS*20], our method assumes that the camera calibration can be performed accurately. However, in challenging cases with, for example, limited camera translation, it may not be possible to obtain accurate camera poses using COLMAP. Such inaccuracies could potentially negatively affect the quality of our results. It would be interesting to address this problem by investigating the possibility of optimizing the camera poses similar to the approach by Kopf et al. [KRH21].

Moreover, our geometric constraints are designed based on the assumption that the scene is static. For moving areas, the geometric constraints are weak and our method, similar to CVD, relies on the learned priors. While our approach produces considerably better results than CVD in dynamic regions, we could benefit from explicitly modeling the motion. An interesting future research would be to address this issue by incorporating the scene flow estimation network with temporal losses, as proposed by Zhang et al. [ZCT*21], into our system. Finally, currently all the test-time optimization approaches [LHS*20, KRH21, ZCT*21], including ours, rely on optical flows to enforce the geometric constraints. Although the forward backward consistency tests and our median depth strategy mitigate the problem caused by inaccurate flows, our method would struggle to produce high-quality results in cases where the optical flows are highly inaccurate. We believe improving the quality of the optical flow through test-time optimization is an interesting avenue for future research.

5. Conclusion

We present a novel test-time optimization approach for estimating geometrically and temporally consistent depth from a monocular

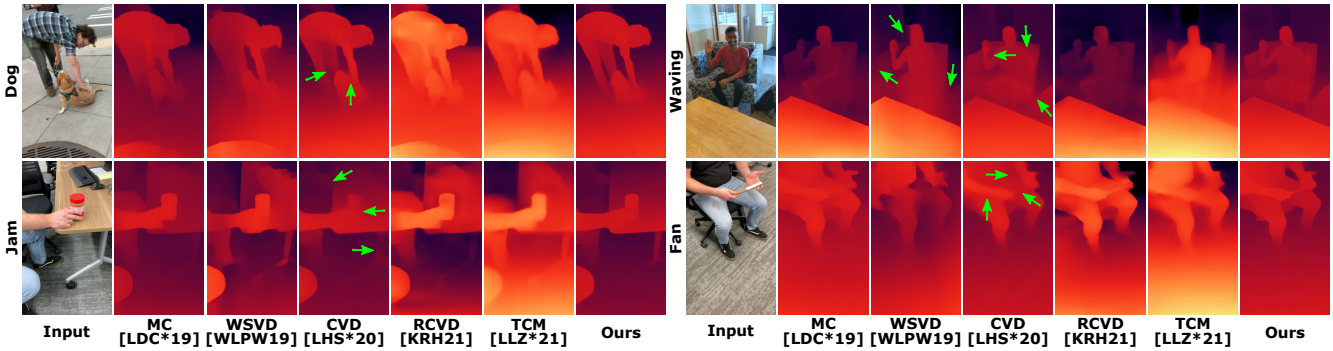

Figure 8: *Comparisons against several state-of-the-art methods on casually captured dynamic scenes. See the full videos in the supplementary video.*

Table 4: *Evaluating the effect of different design choices on the TUM RGB-D dataset.*

	Error Metric ↓				Accuracy Metric ↑		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$
Ours w/o pseudo loss	0.4864	0.2474	0.3268	0.5479	0.5026	0.6593	0.7786
Ours w/o consistency loss	0.1747	0.0908	0.1554	0.2638	0.7776	0.9099	0.9577
Ours using disparity loss	0.1891	0.0906	0.1639	0.2830	0.7532	0.8939	0.9505
Ours w/o median operation	0.1613	0.0438	0.1538	0.2221	0.7687	0.9224	0.9741
Ours w/o confidence map	0.1398	0.0439	0.1325	0.1962	0.8271	0.9358	0.9740
Ours $\lambda = 0.1$	0.1652	0.0536	0.1623	0.2133	0.7942	0.9325	0.9531
Ours $\lambda = 0.9$	0.2114	0.0802	0.1918	0.2405	0.7507	0.8535	0.9248
Ours	**0.1346**	**0.0345**	**0.1240**	**0.1879**	**0.8317**	**0.9414**	**0.9788**

video. Specifically, we use an existing single image depth estimation network and optimize its parameters on the test example at hand. Our main contribution is to compute pseudo reference depth by matching optical flow and depth-reprojection displacements. We propose a strategy to discard the erroneous pseudo reference depth maps computed from pairs of images to obtain a per-frame pseudo reference depth. We also propose a simple method to compute a confidence map for each per-frame pseudo depth. Our test-time loss ensures that the estimated depth and the pseudo reference depth at each frame are similar and the estimated depth for neighboring frames are consistent. We demonstrate the efficiency and effectiveness of our approach through comparisons against the state-of-the-art techniques both visually and numerically.

6. Acknowledgement

We thank the reviewers for their insightful comments. We also thank Brennen Taylor for capturing the input sequences.

References

[CFYD16] CHEN W., FU Z., YANG D., DENG J.: Single-image depth perception in the wild. In *International Conference on Neural Information Processing Systems (NIPS)* (2016).

[CPMA19] CASSER V., PIRK S., MAHJOURIAN R., ANGELOVA A.: Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Association for the Advancement of Artificial Intelligence* (2019).

[CQD19] CHEN W., QIAN S., DENG J.: Learning single-image depth from videos using quality assessment networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Los Alamitos, CA, USA, jun 2019), IEEE Computer Society, pp. 5597–5606.

[DPH*20] DAI Q., PATII V., HECKER S., DAI D., GOOL L., SCHINDLER K.: Self-supervised object motion and depth estimation from video. *The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (06 2020), 4326–4334.

[EF14] EIGEN D., FERGUS R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *International Conference on Neural Information Processing Systems (NIPS)* (2014).

[EPF15] EIGEN D., PUHRSCH C., FERGUS R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *International Conference on Computer Vision (ICCV)* (2015).

[FGW*18] FU H., GONG M., WANG C., BATMANGHELICH K., TAO D.: Deep Ordinal Regression Network for Monocular Depth Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).

[GAFB19] GODARD C., AODHA O. M., FIRMAN M., BROSTOW G.: Digging into self-supervised monocular depth estimation. In *International Conference on Computer Vision (ICCV)* (2019), pp. 3827–3837.

[GGAM14] GUPTA S., GIRSHICK R., ARBELAEZ P., MALIK J.: Learning rich features from RGB-D images for object detection and segmentation. In *The European Conference on Computer Vision (ECCV)* (2014).

[GLJA19] GORDON A., LI H., JONSCHKOWSKI R., ANGELOVA A.: Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *International Conference on Computer Vision (ICCV)* (Los Alamitos, CA, USA, nov 2019), IEEE Computer Society, pp. 8976–8985.

[GLU12] GEIGER A., LENZ P., URTASUN R.: Are we ready for au-

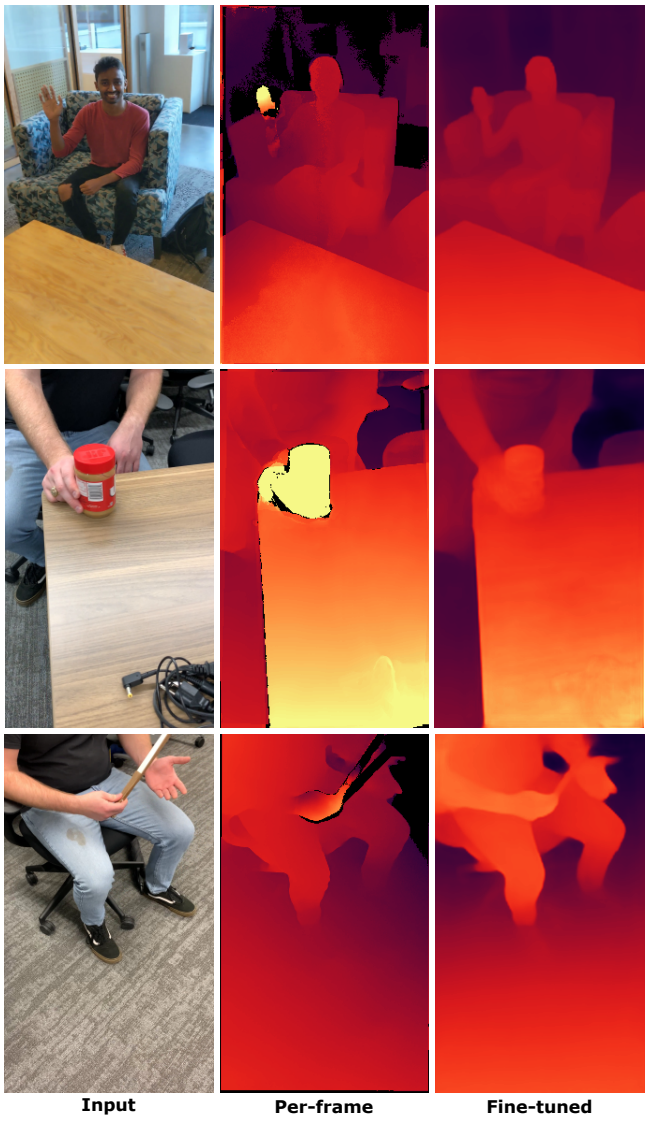

Figure 9: *We show the impact of our test-time optimization process. Per-frame pseudo reference depths suffer from severe artifacts in the moving regions. Our test-time optimization is highly effective in improving the results and producing consistent depth maps.*

tonomous driving? the kitti vision benchmark suite. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2012).

[GLY*18] GUO X., LI H., YI S., REN J., WANG X.: Learning monocular depth by distilling cross-domain stereo networks. In *The European Conference on Computer Vision (ECCV)* (2018), pp. 484–500.

[GMB17] GODARD C., MAC AODHA O., BROSTOW G. J.: Unsupervised monocular depth estimation with left-right consistency. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).

[HZ04] HARTLEY R. I., ZISSERMAN A.: *Multiple View Geometry in Computer Vision*, second ed. Cambridge University Press, ISBN: 0521540518, 2004.

[IMS*17] ILG E., MAYER N., SAIKIA T., KEUPER M., DOSOVITSKIY A., BROX T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 1647–1655.

[KB15] KINGMA D., BA J.: Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)* (2015).

[KRH21] KOPF J., RONG X., HUANG J.-B.: Robust consistent video depth estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2021).

[LDC*19] LI Z., DEKEL T., COLE F., TUCKER R., SNAVELY N., LIU C., FREEMAN W. T.: Learning the depths of moving people by watching frozen people. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).

[LGJA09] LIU F., GLEICHER M., JIN H., AGARWALA A.: Content-preserving warps for 3d video stabilization. *ACM Transcations on Graphics (Proceedings of ACM SIGGRAPH) 28*, 3 (2009).

[LGK*19] LIU C., GU J., KIM K., NARASIMHAN S., KAUTZ J.: Neural rgb®d sensing: Depth and uncertainty from a video camera. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 10978–10987.

[LHS*20] LUO X., HUANG J., SZELISKI R., MATZEN K., KOPF J.: Consistent video depth estimation. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 39*, 4 (2020).

[LLZ*21] LI S., LUO Y., ZHU Y., ZHAO X., LI Y., SHAN Y.: Enforcing temporal consistency in video depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops* (2021).

[LNSW21] LI Z., NIKLAUS S., SNAVELY N., WANG O.: Neural scene flow fields for space-time view synthesis of dynamic scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2021), pp. 6498–6508.

[LRB*16] LAINA I., RUPPRECHT C., BELAGIANNIS V., TOMBARI F., NAVAB N.: Deeper depth prediction with fully convolutional residual networks. In *International Conference on 3D Vision (3DV)* (2016), IEEE, pp. 239–248.

[LS18] LI Z., SNAVELY N.: Megadepth: Learning single-view depth prediction from internet photos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).

[LSL15] LIU F., SHEN C., LIN G.: Deep convolutional neural fields for depth estimation from a single image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015).

[MIH*16] MAYER N., ILG E., HÄUSSER P., FISCHER P., CREMERS D., DOSOVITSKIY A., BROX T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 4040–4048.

[MST*20] MILDENHALL B., SRINIVASAN P. P., TANCIK M., BARRON J. T., RAMAMOORTHI R., NG R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In *The European Conference on Computer Vision (ECCV)* (2020).

[NSF12] NATHAN SILBERMAN DEREK HOIEM P. K., FERGUS R.: Indoor segmentation and support inference from rgbd images. In *The European Conference on Computer Vision (ECCV)* (2012).

[PGDG20] PATIL V., GANSBEKE W. V., DAI D., GOOL L. V.: Don't forget the past: Recurrent depth estimation from monocular video. *IEEE Robotics and Automation Letters 5* (2020), 6813–6820.

[PSB*21] PARK K., SINHA U., BARRON J. T., BOUAZIZ S., GOLDMAN D. B., SEITZ S. M., MARTIN-BRUALLA R.: Nerfies: Deformable neural radiance fields. *International Conference on Computer Vision (ICCV)* (2021).

[PZ17] PENNER E., ZHANG L.: Soft 3d reconstruction for view synthesis. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 36*, 6 (2017).

[QLL*18] QI X., LIAO R., LIU Z., URTASUN R., JIA J.: Geonet: Geometric neural network for joint depth and surface normal estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), pp. 283–291.

[QLL*20] QI X., LIU Z., LIAO R., TORR P. H., URTASUN R., JIA J.: Geonet++: Iterative geometric neural network with edge-aware refinement for joint depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2020).

[QLW*17] QI C. R., LIU W., WU C., SU H., GUIBAS L. J.: Frustum pointnets for 3d object detection from rgb-d data. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).

[RBK21] RANFTL R., BOCHKOVSKIY A., KOLTUN V.: Vision transformers for dense prediction. *ArXiv preprint* (2021).

[RJB*19] RANJAN A., JAMPANI V., BALLES L., SUN D., KIM K., WULFF J., BLACK M. J.: Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).

[SBC12] STURM J., BURGARD W., CREMERS D.: Evaluating egomotion and structure-from-motion approaches using the TUM RGB-D benchmark. In *Proc. of the Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RJS International Conference on Intelligent Robot Systems (IROS)* (Oct. 2012).

[SF16] SCHÖNBERGER J. L., FRAHM J.-M.: Structure-from-motion revisited. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[SLX15] SONG S., LICHTENBERG S. P., XIAO J.: Sun rgb-d: A rgb-d scene understanding benchmark suite. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015).

[SZPF16] SCHÖNBERGER J. L., ZHENG E., POLLEFEYS M., FRAHM J.-M.: Pixelwise view selection for unstructured multi-view stereo. In *The European Conference on Computer Vision (ECCV)* (2016).

[TS20] TUCKER R., SNAVELY N.: Single-view view synthesis with multiplane images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2020).

[VKB*18] VALENTIN J., KOWDLE A., BARRON J. T., WADHWA N., DZITSIUK M., SCHOENBERG M. J., VERMA V., CSASZAR A., TURNER E. L., DRYANOVSKI I., AFONSO J., PASCOAL J., TSOTSOS K. N. J., LEUNG M. A., SCHMIDT M., GULERYUZ O. G., KHAMIS S., TANKOVICH V., FANELLO S., IZADI S., RHEMANN C.: Depth from motion for smartphone ar. *ACM Transactions on Graphics* (2018).

[VRS*17] VIJAYANARASIMHAN S., RICCO S., SCHMID C., SUKTHANKAR R., FRAGKIADAKI K.: Sfm-net: Learning of structure and motion from video. *ArXiv abs/1704.07804* (2017).

[WLPW19] WANG C., LUCEY S., PERAZZI F., WANG O.: Web stereo video supervision for depth prediction from dynamic scenes. In *International Conference on 3D Vision (3DV)* (2019).

[WPF19] WANG R., PIZER S., FRAHM J.-M.: Recurrent neural network for (un-)supervised learning of monocular video visual odometry and depth. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 5550–5559.

[YKG*20] YOON J. S., KIM K., GALLO O., PARK H. S., KAUTZ J.: Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2020).

[YS18] YIN Z., SHI J.: Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).

[ZBSL17] ZHOU T., BROWN M., SNAVELY N., LOWE D. G.: Unsupervised learning of depth and ego-motion from video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).

[ZCT*21] ZHANG Z., COLE F., TUCKER R., FREEMAN W. T., DEKEL T.: Consistent depth of moving objects in video. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)* (2021).

[ZJ20] ZACHARY T., JIA D.: Deepv2d: Video to depth with differentiable structure from motion. *International Conference on Learning Representations (ICLR)* (2020).

[ZLH18] ZOU Y., LUO Z., HUANG J.-B.: Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *The European Conference on Computer Vision (ECCV)* (2018).

[ZSL*19] ZHANG H., SHEN C., LI Y., CAO Y., LIU Y., YAN Y.: Exploiting temporal consistency for real-time video depth estimation. *International Conference on Computer Vision (ICCV)* (2019), 1725–1734.

[ZTF*18] ZHOU T., TUCKER R., FLYNN J., FYFFE G., SNAVELY N.: Stereo magnification: Learning view synthesis using multiplane images. In *SIGGRAPH* (2018).

Appendix A: Derivation of Pseudo Reference Depth

Here, we derive the closed form solution to Eq. 2. To do this, we first find the reprojected pixel $q_{i\to j}$ that minimizes the objective in Eq, 2. Once this pixel is found, we can obtain the optimal depth corresponding to this reprojected pixel. Since $q_{i\to j}$ is the depth reprojected pixel q, as shown in Fig. 2, it can only lie on the epipolar line corresponding to pixel q. This epipolar can be easily calculated [HZ04] and represented using the epipole e_j and a unit direction vector $\vec{v}_e$. To calculate the optimal reprojected pixel $q^*_{i\to j}$, we find the point on the epipolar line with smallest distance to pixel p. This can be obtained as:

$$q^*_{i\to j} = e_j + ((p - e_j)\cdot \vec{v}_e)\cdot \vec{v}_e, \tag{10}$$

This point is projected from a 3D point that lies on the line originating from o_i in unit direction $\vec{v}_q$. To obtain this point and consequently the optimal depth, we intersect this line with the one originating from o_i in unit direction $\vec{v}_o = (q^*_{i\to j} - o_j)/\|q^*_{i\to j} - o_j\|$. Representing these two lines as $l = o_i + t\vec{v}_q$ and $m = o_j + s\vec{v}_o$, we can use the fact that at the intersection point the dot product of the vector connecting these two lines with the unit direction vector of each line should be equal to zero. This provides us with the following system of equations:

$$\begin{aligned}(o_i - o_j)\cdot \vec{v}_q + t - s(\vec{v}_o\cdot \vec{v}_q) = 0\\ (o_i - o_j)\cdot \vec{v}_o + t(\vec{v}_q\cdot \vec{v}_o) - s = 0\end{aligned} \tag{11}$$

where t and s are the two unknowns. The solution to t, scaled by $\|cos(\theta)\|$, is the optimal depth given in Eq. 3.